Collaborative Virtual Sensorweb Infrastructure: Architecture and Implementation

Prasanta Bose, Neal Hurlburt, Ankur Somani: Lockheed Martin, ATC
Peter Fox: UCAR

Abstract. Geoscience and space science research involving exploitation of insitu sensor networks and remote observing instruments offer an unprecedented opportunity for coordinated observation and science data processing critical to science understanding and timely prediction of climate and (earth and space) weather changes. Current efforts in specialized virtual sensors or observatories targeting the atmosphere, ocean, carbon management, space weather and etc. domains raise the common challenge to create a collaborative sensorweb infrastructure that coordinates sensor resources and data products from individual observatories for effective and robust science processing and timely weather prediction. The challenges to creating such an infrastructure include model-based (model-predictive) integration of processing services and data spanning multiple sources, intelligent push of the data in an efficient, timely manner and closed-loop decentralized management and reconfiguration of the sensing resources to meet dynamic science and prediction needs. Semantic descriptions (via OWL-S, SensorML) of services and resources via ontologies and semantic markup languages provide a basis for the development of middleware infrastructure enabling intelligent and flexible (goal-driven) management of the services and physical resources. Our NASA funded Collaborative Sun-Earth Connector (CoSEC) project and Virtual Sensor Web Infrastructure for Collaborative Sensing (VSICS) project has been developing a common scaleable architecture and agent-based (event-driven) coordination (e.g. workflow management) and control (e.g. planning and scheduling) services for such an infrastructure for application to the space science and the geosciences domain respectively. This paper will present features of the architecture, and current implementation of the infrastructure applied to collaborative sensorwebs.

Index Terms:* *Sensorweb, Middleware, Resource Management, Weather Monitoring, Semantic description, Collaborative Sensing

I. Introduction

Geoscience and space science research involving exploitation of insitu sensor networks and remote observing instruments offer an unprecedented opportunity for coordinated observation and science data processing critical to understanding and timely prediction of climate and (earth and space) weather changes. Current efforts [5, 6, 10] in specialized virtual sensors or observatories targeting the atmosphere, ocean, carbon management, space weather and earth-science domains raise the challenge to create a common collaborative sensorweb infrastructure that coordinates sensor resources and data products from individual observatories for effective and robust science processing and timely weather prediction.

Current sensor web infrastructures (exemplified by in-situ sensor networks and remote space and air-based sensing platforms) fail to dynamically adapt and provide robust performance in context of above tasks. The limitations arise from inflexibility with respect to exploiting new sensing platforms. Current sensorweb architectures and deployments (as in sensor networks) are statically configured at design and deployment time. They are configured and optimized based on specific assumptions about the operating environment and tasks. Run-time evolution of such networks via addition of new platforms or via changes in the services provided by the platforms is very difficult and costly in such frameworks. Further, sensorwebs fail to reason about non-functional attributes of resources and services to meet a user's or application's utilities and constraints in context of changing environment and mission context. Knowledge about these non-functional attributes and reasoning about such attributes to make resource allocation decisions is either very limited or nonexistent. Without the awareness of these attributes in regards to the resources and their settings (or their degrees of freedom), it makes it impossible to reason about a user's utility and constraint requirements and make fair allocation of resources or make tradeoffs that maximizes the system utility in a given context. These attributes include sensor error and reliability, uncertainty of their collected data, computational resource (e.g. computing power and buffer) limitations, user priorities, network errors, sensing precision, accuracy, etc. Additionally, existing sensorwebs fail to dynamically manage resource settings in response to changing demands and context. Resource settings are generally statically configured. They are set in the implementation (C code, Java code, etc.) and are set to respond to the demand and

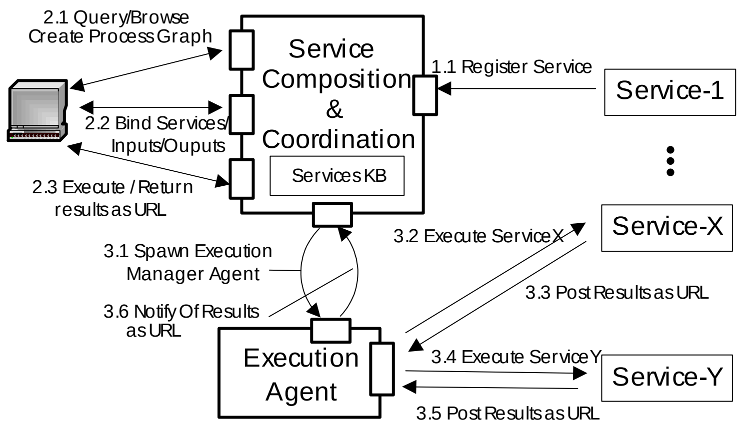


Figure 1. Component Interactions in COSEC.

context of the sensor network as it is during the time of its instantiation. However, if demands on the network or the context in which the sensors and services are to be used changes, there is no systematic way to autonomously adapt assignment of services and resources to take these changes into account. This paper introduces the Virtual Sensor Web Infrastructure for Collaborative Science (VSICS) Architecture currently being developed to create a virtual sensor web for sustained coordination of (numerical and distributed) model-based processing, closed-loop resource allocation, and observation planning. The virtual architecture enables one to create an overlay architecture that enables both hierarchical management and scaling in context of sensorwebs consisting of heterogeneous resources or nodes.

II. PRIOR WORK - *COLLABORATIVE SUN-EARTH CONNECTOR (COSEC)*

The goals of the CoSEC project are to offer a set of capabilities for off-line analysis of scientific data via goal-driven composition of analysis services developed by users. The key features of the CoSEC architecture for distributed data analysis and assimilation are:

Multi instrument, multi-mission, multi-PI collaborative process driven asynchronous data assimilation. The architecture supports workflow-oriented collaborative data assimilation, plan representation, and composition of processing services, data and resources to match roles and constraints.

Plug-and-play, scaleable architecture with security attributes. CoSEC implements registration, discovery and semantic integration of services ranging from application-specific ones to execution and resource management services (for integration with the grid). It also *ensures robust and secure data analysis* in a distributed context. Scalability is achieved via decentralized coordination of modular virtual machines (or servers) cooperating to perform assimilation.

Resource management for collaborative assimilation. CoSEC allows design-time binding of service instance nodes with the user application workflow tasks to meet current demands. This capability allows multiple (mirror) sites to register and compete to dynamically bind and execute user tasks. Moreover, CoSEC allows dynamic configuration and resource management for durable quality of service over the life of a long-term composite processing task. Figure 1 depicts the current CoSEC architecture and the key interactions between its components. CoSEC treats LWS data as a set of distributed services, including standard directory lookup and data extraction and more complex tasks like calibration, windowing, and data compression. Figure 2. Process-based workflow model in COSEC for science data analysis. The workflow modeling and execution environment is based Ptolemy [11]. The Ptolemy extensions in COSEC enables services to be searched for based on high-level descriptions [3,4] and be linked together to form composite services with the results of one service feeding directly into another (Figure 2).

VSICS extends COSEC to support online and offline adaptation of instrument and data-analysis operations. Offline optimization requires repositories to tag data products with the workflow that generated them and the performance and data quality metrics used by scientists to evaluate those products. This is already being incorporated into CoSEC for capturing data analysis operations. This enables VSICS to approximate the

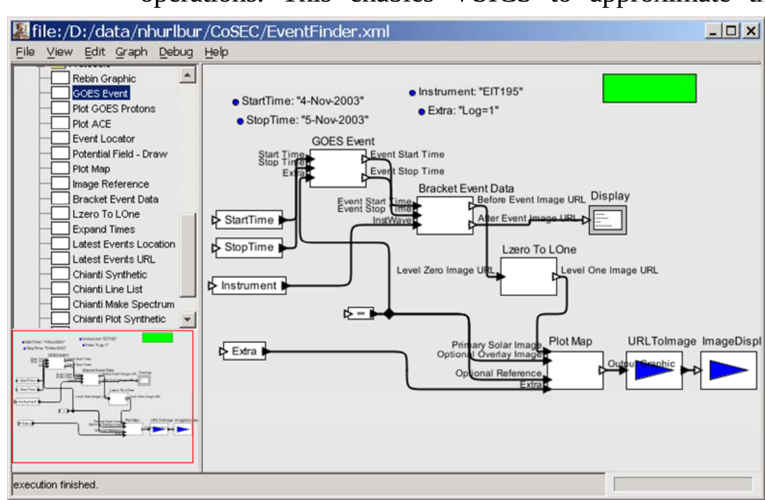


Figure 2. Process-based workflow model in COSEC for science data analysis.

value of different process networks for observing the same phenomena and represent the sensitivity of a particular model to changes in scientific goals or solar phenomena. VSICS can apply the workflows generated by these simulations to future observing tasks, and iteratively improve its performance models over time. Online optimization allows VSICS to monitor the state of workflow elements and dynamically reorder or reassign tasks as science objectives change and resource attributes change due to contention, failure and other dynamics. A key objective of VSICS is extending the capabilities of COSEC with autonomous resource binding and adaptation for robust performance. Currently, a human controller must directly communicate with the service, requiring the user to have low level knowledge of the resource. Also, non-functional information on resources does not exists in the current implementation. As described in the previous section, this would be highly valuable to the end user. The data received from a service or sensor is currently served to one central repository which a user must query with detailed information to extract the appropriate information. As with the user needing to have a good understanding of the resource model to communicate and control or task it, the user would need to have thorough knowledge regarding what they are looking for in order to effectively query all stored results. Clearly, this adds an undesirable burden on the end-user.

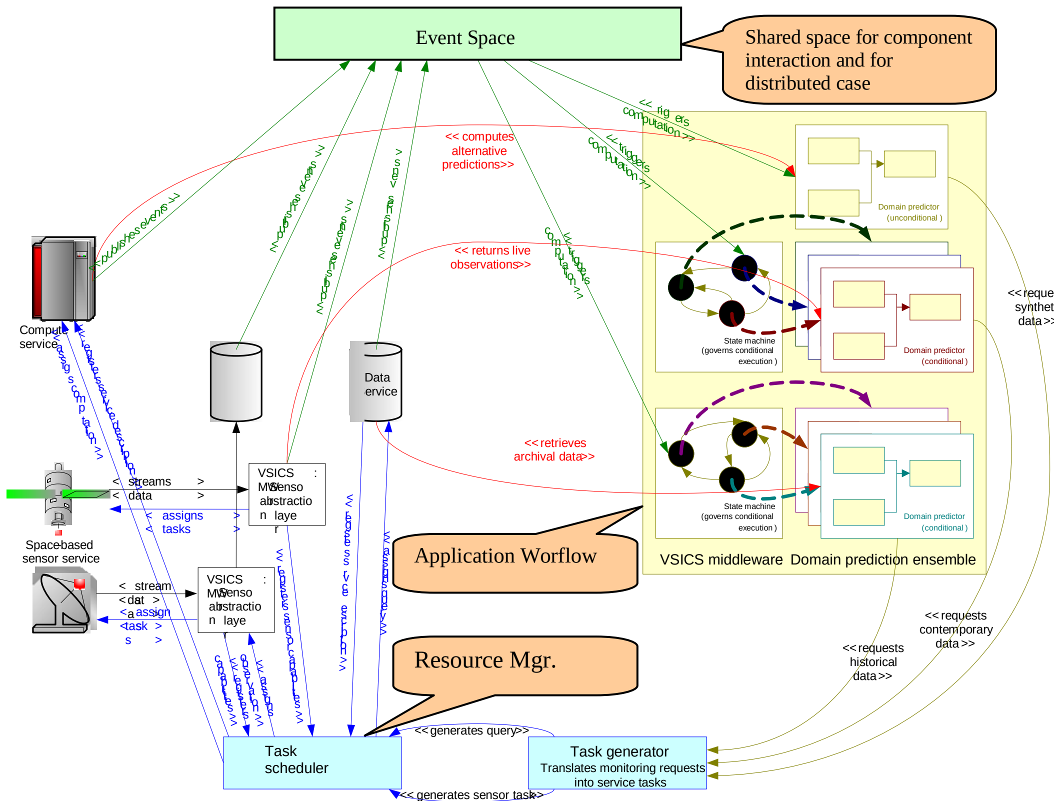


Figure 3. VSICS Architecture -- Centralized Version

III. VSICS Approach

The overarching goal of VSICS is to bridge the gap between the end-user applications and the resource-management for autonomous adaptive management of resources and services leading to effective utilization of the sensorweb resources and robust performance in context of a collaborative task such as weather monitoring and prediction, science processing and analysis and other tasks.

A. Concept of Operations

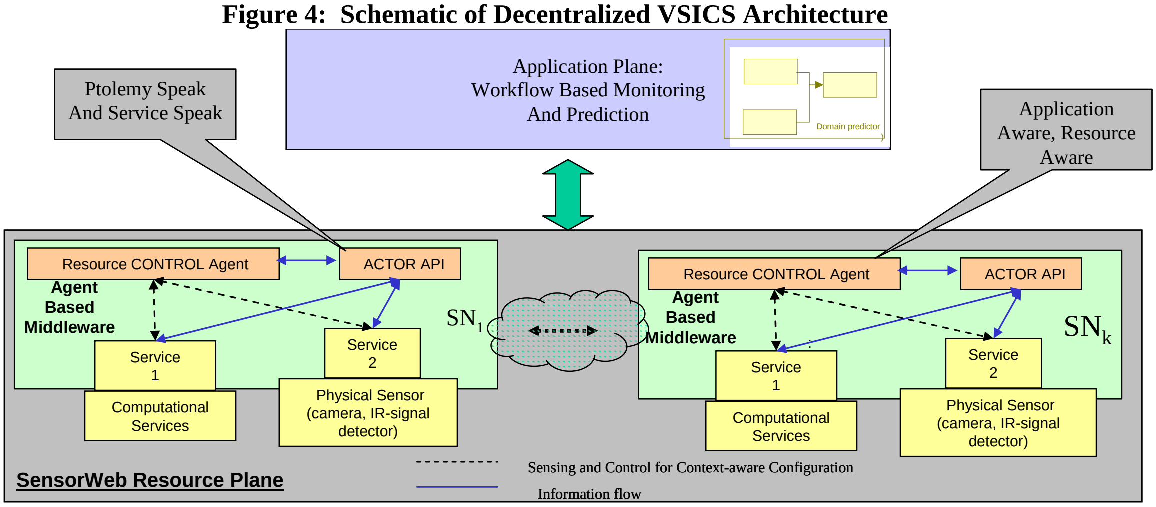
The VSICS system is based on a multi-plane architecture – an application plane and a resource plane. In broad terms, the application plane can also be viewed as the user space, where prediction of events and science processes occur. The resource plane is where the sensor web nodes reside and are providing services. Essentially, the nodes in the resource plane are utilized by the users (or applications) in the application plane to obtain access to data or services.

There are multiple options to coordinate the control and management of the resources. The VSICS architecture attempts to push the application constraints and run-time control into the resource management plane. The two differing architectures explored here are a centralized architecture as shown in figure 3 and a decentralized architecture as shown in figure 4.

The application plane, in the architecture shown in figure 3, is essentially being comprised of everything in the yellow box on the right labeled as “Consumer science product workflows”. The application plane consists of the user’s science processes, as well as prediction (such as domain prediction) processes. The resource plane, as shown by the bottom and left portion of this figure, includes the task generator, task scheduler, and the sensor nodes themselves. The sensor nodes provide a set of services. The task generator and task scheduler constitute the resource plane and perform open-loop and closed-loop tasking of the sensors consistent with the resource constraints and application requests. The event shared space (green colored box in the figure) enables event-driven coordination of cross-plane components

Figure 4 shows the decentralized view of the VSICS architecture. The application plane is labeled as the purple box on the top of the figure. As in the centralized design, the application plane comprises of workflows instantiated for monitoring and prediction tasks. The resource plane consists of networked sensor-web nodes (SN_1 to SN_k). Each sensor web node has a resource control agent (with capabilities similar to the task generator and task scheduler in the centralized view) that is application aware and resource aware. Application awareness stem from the node’s ability to exploit declarative representation of constraints and requests posted by process in the application lane. Self-awareness is enabled by exploiting representation of the system state of the resources that it can access. Another major component in the resource plane is the Actor. Actors serve an important role in providing a bridge between the science processing/monitoring workflows and services in the resource plane.

Figure 4: Schematic of Decentralized VSICS Architecture



By pushing the resource management responsibility into the resource plane, and developing autonomous sensorweb nodes that perform context-aware control (via the resource control agents) to provide the information services (via the Actor API), the user (workflow) in the application plane can focus on creating a science processing workflow and/or (cooperative) weather analysis rather than worrying about how to collaborate with other users to share information services and resources, perform tasking and scheduling of resources to meet joint needs. This allows for opportunistic utilization and configuration, based on what the user’s needs or task (weather monitoring, etc) specific information needs. The resource control agent exploits awareness of the managed sensor services, their various non-functional state attributes, and awareness of the QoS requirement of the applications and their information needs to make resource assignment and control decisions at run-time. When such resource control decisions are managed in a centralized manner – semantic descriptions [2, 13] of the sensor-web node services and its states are leveraged to make intelligent decision making (via reasoning processes) that attempt to mazimize certain objectives (e.g. utilization of the resource. In a decentralized decision context – semantic descriptions of services are used to advertise capabilities and exploited by the resource control agent for cooperative teamwork to meet application needs. Semantic descriptions are also used in VSICS to specify policies that refer to the services and gets used at run-time for adaptive configuration and allocation of sensor web resources.

B. Key Elements

The key elements of the VSICS infrastructure relevant to adaptive resource management (that synergistically work with the application plane management processes) include mechanisms for describing sensorweb node services, (utility optimization or deterministic policy) based reasoning agents, and mechanisms for distributed signaling and coordination for control and data flow.

For smart adaptive resource management – VSICS considers sensor-web node services to be not only providing and processing of application specific information needs and processing but also services for managing the resources at the node such as power, physical sensor, etc. As discussed in the previous sections, semantic descriptions of the services play an important role for development of flexible and evolvable reasoning based allocation of resources.

Here we briefly describe how description of services are used in meta policies for policy-driven management of services. Meta-policies in VSICS have the following general syntax: <Trigger> + <Condition1.. k> ➔ <Action>. In other words: on *trigger* if *conditions* do *action*. The "trigger" part of the policy captures the environmental awareness, or the event/state that triggers this rule. This could include cache usage and failure rates. The "condition" part models self awareness, or the service capability and conditions on it's attributes that must be true. This could also include application specific goals and resources constraints. The "action" part specifies the resource management service settings/configuration to perform if the condition has been determined to be true. It is to be noted, that COSEC focuses primarily on the application plane and automated binding of information processing services to the workflow via semantic descriptions.

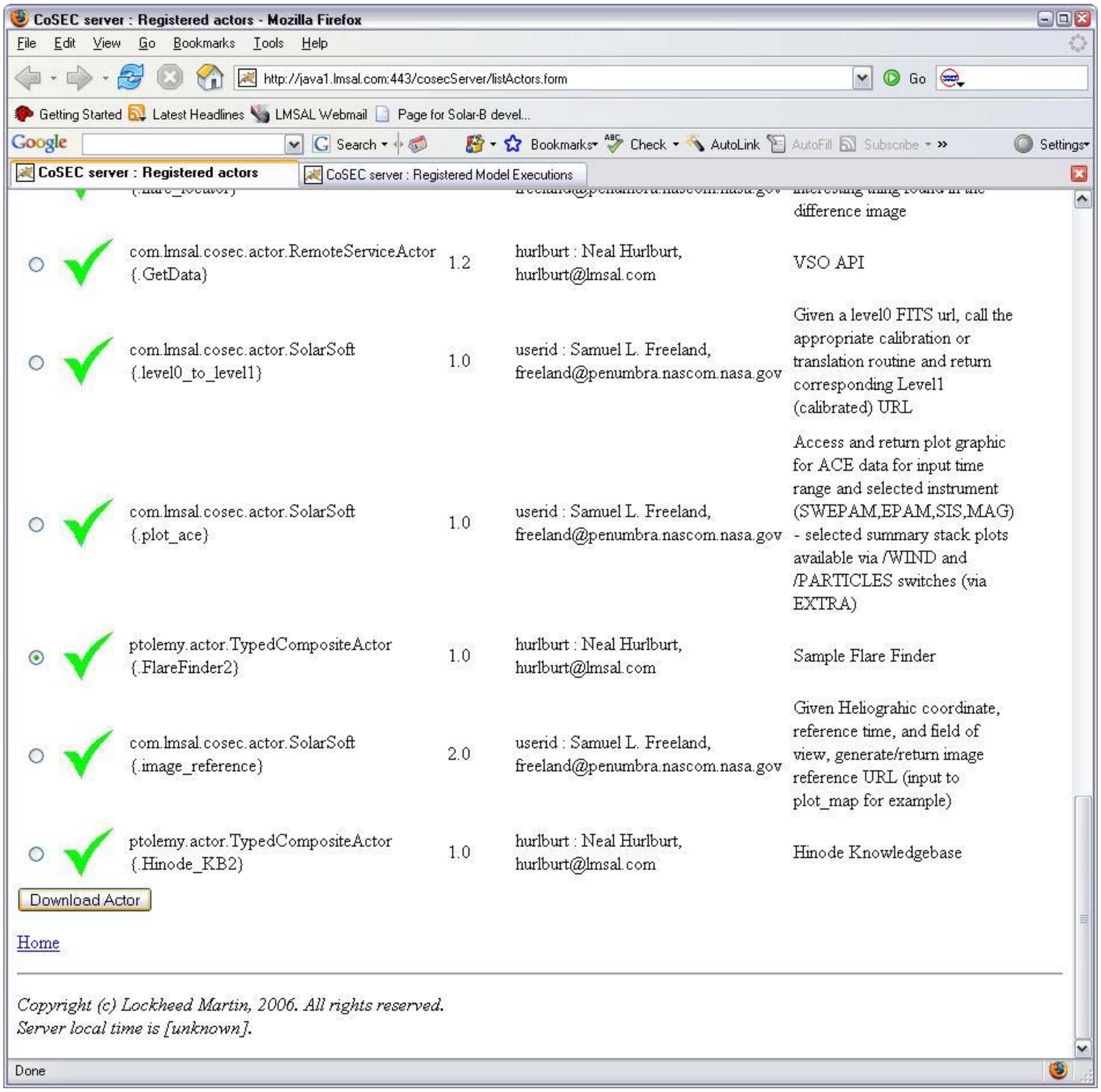


Figure 5. Registered Services in COSEC/VSICS Example.

IV. Initial Results

A. CoSEC (Collaborative Sun-Earth Connector)

As mentioned previously, the current implementation extends the CoSEC project and leverages off application plane and enable the user to perform workflow creation. Currently, CoSEC provides a workflow client and server processes to provide a central repository for services and product results. A user is able to register, or essentially display and make available, an actor that performs some sort of function (Fiugure 5). Another user may obtain this actor and import it into the CoSEC client. Once there, the user can add this actor to a workflow, or if the actor already consists of a complete workflow, the user can simply execute the workflow. Once the workflow has executed, and the specified service calls have been made, the CoSEC client will display the results from the workflow and any services called . The execution of this actor or workflow is also registered and stored in the CoSEC central repository (Figure 5). In addition, the results displayed to the user in the CoSEC client is also registered and stored in the CoSEC central repository . Any of the items in the CoSEC central repository (registered actors, registered execution, or registered result) can be viewed via a web page. A user can later access the central repository and search for a specific result if he doesn't want to or intend on running a workflow and actor again.

B. VSICS Use Cases: Scenarios

The spiral development and evaluation of the VSICS framework is guided by a set of real-world use cases from forest-fire and volcanic eruptions monitoring domain. We briefly discuss the elements of the forrest-fire scenario to provide an understanding of the kind of services and dynamics (from requirements standpoint) that VSICS need to manage.

Background. Lightning in the Alaska-Yukon border area triggered numerous large forest fires in the summer of 2004, carrying smoke and other aerosols high into the atmosphere. The upwelling of particulate matter from the fires could be seen as a red plume moving across North America. For much of the summer, the particles remained high in the atmosphere and did not settle over populated areas. As the aerosols were advected into the southern US, the meteorological conditions changed: A cold front developed off the Eastern seaboard, and the dynamics of the front forced a major portion of the plume to descend to the surface and impact the air quality along the Eastern US. Episodes such as this have ramifications for human and ecosystem health and productivity on both short and longer-term time scales.

Analysis. This particular lightning/wildfire event scenario leads to a specific configuration of the data analysis and model processes needed to make effective predictions and assessments of wildfire development and movement. Some of these requirements in turn lead to subscription for detected/observed events that get published by data sources (event detection services are part of the Earth/Space data ingest/data systems). These data are placed into a shared event database together with a binding of the propagation specific tasks to computational processes (simulation services). Often, depending on climatic conditions (e.g. seasonal variations or wind direction reversals) it is possible that some models that are triggered by same event may have very different results and thus initiative distinctively different follow-on tasks (see http://www.ucar.edu/research/pollution/). This scenario is particularly suited to demonstrate the capabilities of VSICS since it operates on multiple time scales. In the near term, it affects the local climate and population close to the actual wildfire event. However, in the middle to long term, it affects populations that are thousands of miles away.

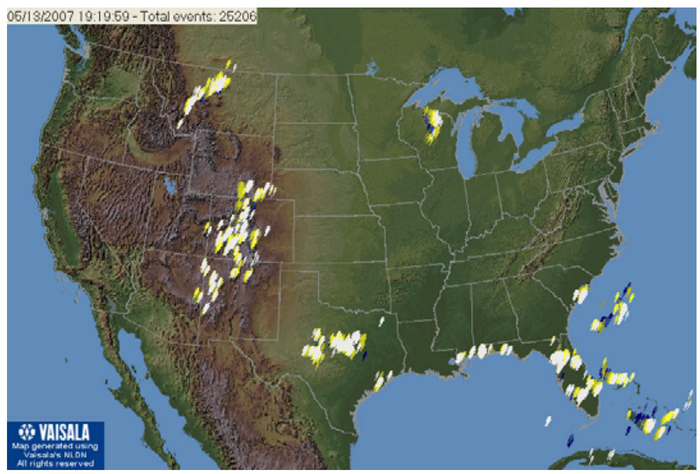


Figure 6. Forrest-fire Activity.

Weather prediction and more specifically air quality prediction can be greatly augmented by information provided by neighboring area volcanic activity monitoring efforts. We would like to highlight the areas of this interaction where the volcanic activity monitoring effort comes to a conclusion that the volcanic activity will have an effect in the immediate future on the weather and air quality in neighboring regions, and subsequently task the weather prediction science data product to utilize the conclusions made by the volcanic activity monitoring science data product to generate a more accurate prediction of impending weather and air quality. A possible addendum to this is the possible situation where predicted weather, such as a cold or warm front, may in fact have an effect on the volcanic activity, and thus the weather prediction science data product will task the volcanic activity monitoring science data product to run utilizing the weather prediction results.

In order to fully develop, deploy, and test the VSICS framework detailed use cases, such as those above, along with physical sensors and their associated services must be available for use. To this end, VSICS need to be provide APIs for integration with existing embedded sensor networks such as the Lightning Detection Network (LDN). In addition, loosely coupled (mobile) sensors (balloons) and models that could be dynamically initiated by an event (e.g. front moving in on north-west coast or change in wind direction invokes new/ different model run/ configurations) will be identified and employed. Further, remote sensing resources, such as those from NASA and NOAA, will be located and made use of.

Many physical sensors and associated services already exist that planned VSICS experiments will leverage off of for the wildfire scenario. Wide band magnetic Direction Finding (DF)/IMPACT sensors exist that have accuracy on the order of 8 km within the nominal range of 250 mi (400 km) of the high gain sensors. The quality of the lightning data also depends upon the detection efficiency of the sensors (documented to average around 70%). Time-of-arrival (TOA) sensors can monitor individual return strokes in a multiple lightning flash only 15 ms apart, discriminating more than 50 strokes per second. The nominal time for locating a strike and displaying it on the video monitor is 0.3 sec. Over water reception can be at ranges up to 1600 mi (2600 km) and over land about 1360 mi (2200 km). The National Lightning Detection Network (NLDN) and the North American Lightning Detection Network (NALDN) provide (with subscription) services for analysis, display, and notification.

V. Conclusion

Current sensor web infrastructures fail to dynamically evolve and adapt at the application plane and at the resource plane in order to provide robust performance in context of weather monitoring and science data collection and processing tasks. The limitations arise from inflexiblity with respect to exploiting new sensing platforms and new services, reason about the current collaborative mission and system context to make optimal resource management decisions. We have presented our initial VSICS architecture and initial results that aim to address such a challenge by exploiting explicit knowledge of the resources and services specified in semantic description languages. Current work is focused on development of the (utility-based) reasoning components and the event based coordination for both hierarchical/centralized and decentralized management

Acknowledgment

We would like to acknowledge the NASA AIST and NASA LWS for their support in development of the COSEC and VSICS Architecture.

References

[1] T. Hoeksema and the SSSC roadmap committee, "2005 SSSC Foundation Roadmap," draft: http://sun.stanford.edu/~todd/roadmap/draft2.html

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American, 284(5):34-43, 2001.

[3] P. Bose, M. Woodward, N. Hurlburt, and S. Freeland, "Information Fusion in the Sensor Web", NASA Earth Science Technology Conference, 2002.

[4] N. E. Hurlburt, Freeland, S.L., Bose, P., and Woodward, M., "Semantic Composition of Distributed Solar Data and Analysis Services for Coordinated E-Science", AAS 200th meeting, Albuquerque, NM, June 2002.

[5] J. Postel and J. Touch, "Network infrastructure", Ian Foster and Carl Kesselman, editors, The Grid: Blueprint for a New Computing Infrastructure, chapter 21, pages 552–562, Morgan Kaufmann Publishers, Inc., 1999.

[6] D. De Roure, N. Jennings, N. Shadbolt, and M. Baker, "Research Agenda for the Semantic Grid: A Future e-Science Infrastructure", Technical Paper, 19 February 2002, http://www.semanticgrid.org/html/semgrid.html

[7] J. Gurman, G. Dimitoglou, F. Hill, S. Wampler, R. Bogart, K. Tian, P. Martens, and A. Davey, "The Virtual Solar Observatory: The Best Laid Schema", SPD meeting 34.203, 2003

[8] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs and Y. Xiong, "Taming Heterogeneity---the Ptolemy Approach," *Proceedings of the IEEE*, 91(2), January, 2003.

[9] B. Ludäscher, Arcot Rajasekar, Joseph Goguen, David Stockwell and Victor Vianu, *NSF ITR (large): Enabling the Science Environment for Ecological Knowledge (SEEK)*, October, 2002, https://www.fastlane.nsf.gov/servlet/showaward?award=0225674.

[10] J. Orcutt, F. Vernon, A. Rajasekar, B. Ludäscher, K. Lindquist, *NSF ITR: Real-Time Data Aware System for*

Earth, Oceanographic, and Environmental Applications (ROADNet), September, 2003.

[11] Edward A. Lee, "Overview of the Ptolemy Project," *Technical Memorandum UCB/ERL M03/25*, July 2, 2003, University of California, Berkeley, CA, 94720, USA.

[12] SENSORML - http://vast.uah.edu/SensorML/